REMARKS

The Office Action dated June 22, 2007 has been received and carefully noted. The following remarks are submitted as a full and complete response thereto. Claims 22-42 are currently pending in the application. Claims 22-42 are respectfully submitted for consideration.

In the Office Action, at page 4, claims 22-25, 27-32, 34-39, 41, and 42 were rejected under 35 §103(a) as being unpatentable over Egbert (U.S. Patent No. 6,407,960) in view of Chieng (U.S. Patent No. 6,035,346). The Office Action took the position that Egbert discloses all of the elements of the claims, with the exception of "when it is determined that the default values are obtained through a microprocessor interface, changing said default values according to data received through the microprocessor interface." The Office Action then cites Chieng as allegedly disclosing this limitation of the claims. The rejection is respectfully traversed for the following reasons.

Claim 22, upon which claims 23-28 are dependent, recites a method for configuring default values of a network device. The method includes determining whether default values are obtained through a memory interface, and when it is determined that the default values are obtained through the memory interface, performing the steps of: determining from a header whether any default value of the network device should be updated, fetching at least one configuration instruction from a memory when

the determining step determines that the network device should be updated, and changing a register default value of said default values corresponding to an interpretation of at least one configuration instruction. When it is determined that the default values are obtained through a microprocessor interface, changing the default values according to data received through the microprocessor interface.

Claim 29, upon which claims 30-35 are dependent, recites a network device having default values that are flexibly configurable. The network device includes a microprocessor interface, a memory interface, and a register file containing the default values for the network device. The memory interface is configured to receive configuration instructions and the network device is configured to interpret the received configuration instructions such that the corresponding values are mapped to corresponding default values of the register file. The network device is configurable to set default values based on data received through either the microprocessor interface and the memory interface.

Claim 36, upon which claims 37-42 are dependent, recites a network device including means for determining whether the default values are obtained through a microprocessor interface or a memory interface, means for determining from a header whether any default value of the network device should be updated, means for fetching at least one configuration instruction from the memory when the determining step

determines that the network device should be updated, means changing a register default value of said default values corresponding to an interpretation of the at least one configuration instruction, and means for changing said default values according to data received through the microprocessor interface. The means for changing the default values according to data received through the microprocessor interface is configured to change the default values when the means for determining whether the default values are obtained through a microprocessor interface or a memory interface determines that the default values are to be obtained through a microprocessor interface.

Therefore, the present invention provides a new approach for chip and network component vendors to provide system integrators a dynamic configuration using a low cost EEPROM. With this approach, system integrators will have flexibility to change the default values of all configurable registers inside a network device, such as switch/hub chip or components on a PC board. A network device will be able to update configuration setting either through the low cost EEPROM or through a microprocessor interface.

As will be discussed below, Egbert and Chieng, whether considered alone or in combination, fail to disclose or suggest all of the elements of the claims, and therefore fail to provide the advantages and features discussed above.

Egbert generally describes a method by the external memory interface 16 of

storing register data values in selected device registers 12. The external memory interface 16 initiates reading of the external memory 14 in step 50 following a detected reset condition on the device 10 and in response to detection of the external memory device 14 by the memory sensor 18. In particular, the external memory interface 16 begins reading the first memory location of the external memory 14, which corresponds to an even-numbered location 26. The address decoder logic 22 reads the register address value from the even memory location 26 in step 52, and determines in step 54 whether the most significant bit of the register address value is set to 1. See column 3, lines 50-56.

Chieng generally describes a method for reprogramming boot instructions in an adapter card in a computer system. Specifically, by allowing a host processor to control an intelligent Peripheral Component Interface device's reset logic, the PCI device's programmable read only memory can be reprogrammed without storing any reprogramming instructions in the PROM itself. During reprogramming operations the host processor commands the PCI device into a reset-and-hold mode. During this time the host processor downloads reprogramming instructions and replacement code into the PCI's random access memory. When released from reset, the PCI device executes the reprogramming code downloaded by the host processor. See abstract of Chieng.

Applicants respectfully submit that the combination of Egbert and Chieng fail to

teach or suggest all of the elements of the presently pending claims. For example, Egbert and Chieng fail to disclose or suggest, at least, "determining from a header whether any default value of the network device should be updated and fetching at least one configuration instruction from a memory when the determining step determines that the network device should be updated," as recited in independent claim 22 and similarly recited in claim 36. According to an example of the present invention, it is continuously determined whether the RESET signal is set to inactive. If the RESET signal is determined to be inactive, the header of the EEPROM is read. Determination and comparison of a key is made respect to the predefined number in the chip. Once the match is found, then instructions are read from the EEPROM and the corresponding register value is updated.

The Office Action, on page 5, alleges that Egbert discloses "determining from a header whether any default value of the network device should be updated and fetching at least one configuration instruction from a memory when the determining step determines that the network device should be updated" Applicants respectfully disagree. Egbert merely discloses reading of register address value from the even memory location 26, and determining of the most significant bit of the address value set to 1. See column 3, lines 55-59. Egbert does not disclose **determining** from a header **whether** a default value **should be updated** or **fetching** configuration instructions from memory when it is

determined that the device **should be updated**. (Emphasis added).

In the response to arguments section, on page 3, the Office Action took the position that Egbert teaches a network device that receives configuration instructions from an external memory device. Further, the Office Action took the position that Egbert teaches reading a header (MSB) from a location in the external memory to determine if any values in the network device are to be updated. Applicants respectfully disagree with the Office Action's position. Claim 1 recites the steps of determining from a header whether any default value of the network device should be updated and fetching at least one configuration instruction from a memory when the determining step determines that the network device should be updated. However, Egbert does not disclose or suggest such determining and fetching steps. Instead, the method of Egbert merely reads a register address value from an external memory. See column 4, lines 34-35. There is no teaching or suggestion in Egbert that any steps of **determining whether** any value of the network device should be **updated**. (Emphasis added). Therefore, Egbert does not teach or suggest, "determining from a header whether any default value of the network device should be updated and fetching at least one configuration instruction from a memory when the determining step determines that the network device should be updated," as recited in independent claim 22 and similarly recited in claim 36. Consequently, Egbert does not disclose or suggest all of the elements of claims 22 and 36. Chieng, as

acknowledged by the Office Action, also fails to disclose or suggest these elements of the claims.

Furthermore, Applicants respectfully submit that the combination of Egbert and Chieng fails to disclose or suggest "wherein the configuration instruction interpreter is configured to interpret the received configuration instructions such that the corresponding values are mapped corresponding default values of the register file," as recited in claim 29. For example, Egbert and Chieng do not teach or suggest that the network device is configured to interpret received configuration instructions as in the present application.

As discussed above, Egbert merely discloses the reading of register address values and respective data values from external memory 14, and storing the register data values in the destination device registers 12. See column 4, 1-10. Egbert does not disclose or suggest that the device is configured to **interpret** received configuration instructions so that the corresponding values are mapped to corresponding default values of the register file. (Emphasis added). Chieng also fails to disclose or suggest this limitation of claim 29. Accordingly, the combination of Egbert and Chieng fails to disclose or suggest, "wherein the configuration instruction interpreter is configured to interpret the received configuration instructions such that the corresponding values are mapped corresponding default values of the register file," as recited in claim 29.

In the response to arguments section, on page 3, the Office Action also took the

position that Egbert teaches the network device including a memory interface that receives configuration instructions and the addresses read from the external memory are mapped to the register file in the network device. The Office Action also took the position that Egbert teaches the network device that reads the internal memory and interprets what registers to reprogram and what data to reprogram the device with. Applicants respectfully disagree with the Office Action's position. Claim 29 recites a network device including a microprocessor, a memory, and a register file containing default values for the network device. Egbert does not disclose or suggest such network device. Rather, Egbert teaches that specifying the destination device registers 12 for storage of respective register data values stored in odd-numbered memory locations 28 within the external memory 14. See column, lines 38-42. Egbert merely discloses that the register values specifying the destination device registers. Therefore, Egbert fails to teach or suggest, at least, "the network device is configured to interpret the received configuration instructions such that the corresponding values are mapped to corresponding default values of the register file, and wherein the network device is configurable to set default values based on data received through either the microprocessor interface and the memory interface," as recited in independent claim 29. As such, Applicants respectfully request that the rejection of claim 29 be withdrawn.

For at least the reasons above, Applicants respectfully assert that the combination

of Egbert and Chieng fails to disclose or suggest all of the elements of independent claims 22, 29, and 36. Claims 23-28, 30-35, and 37-42 are dependent upon claims 22, 29, and 36, respectively. Therefore, claims 23-28, 30-35, and 37-42 should be allowed for at least their dependence upon claims 22, 29, and 36, and for the specific limitations recited therein.

In the Office Action, at page 8, claims 26, 33, and 40 were rejected under 35 §103(a) as being unpatentable over Egbert (U.S. Patent No. 6,407,960) in view of Chieng (U.S. Patent No. 6,035,346), and further in view of Gates et al. (U.S. Patent No. 5,727,207 hereinafter "Gates"). The Office Action took the position that Egbert and Chieng disclose all of the elements of the claims, with the exception of "determining a key value from the header and comparing the key value with a magic number predefined inside the network device to determine whether any default value of the network device should be updated." The Office Action then cited Gates as allegedly disclosing this limitation of the claims. The rejection is respectfully traversed for the following reasons.

Dependent claims 26, 33, and 40 depend from independent claims 22, 29, and 36, respectively. Claim 26 recites "said step of determining from the header whether any default value of the network device should be updated comprises determining a key value from said header and comparing said key value with a number pre-defined inside network device to determine whether any default value of the network device should be updated."

Claim 33 recites "wherein the network device is configured to receive a header from the memory interface containing a key value from and configured to compare said key value with a pre-defined number to determine whether any default value of said default values should be updated." Claim 40 recites "wherein said means for determining from the header whether any default value of the network device should be updated comprises means for determining a key value from said header and means for comparing said key value with a number pre-defined inside network device to determine whether any default value of the network device should be updated." Gates fails to cure the deficiencies in Egbert and Chieng, and further fails to disclose the elements of the dependent claims.

Gates generally describes a method and apparatus for automatically loading configuration data on reset into a host adapter integrated circuit. Specifically, configuration data indicative of interface requirements for interfacing to a host adapter card are automatically serially loaded on reset from an external device on the card into host adapter integrated circuit on the card. A driver program can then read the configuration data from the host adapter integrated circuit and thereby determine how to interface with the host adapter card. See abstract of Gates.

As discussed above, the combination of Egbert and Chieng fails to disclose or suggest all of the elements of independent claims 22, 29, and 36. Moreover, Gates does not cure the deficiencies in Egbert and Chieng with respect to independent claims 22, 29,

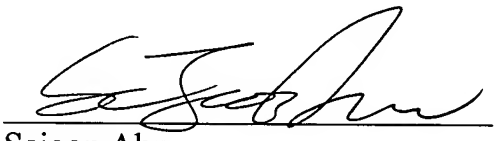
and 36. Gates's method does not determine a key value from a header and compare the key value with a predefined in the network device. Gates fails to disclose or suggest such determination and comparison steps. As such, the combination of Egbert, Chieng and Gates also fails to disclose or suggest all of the elements of claims 26, 33, and 40. Additionally, claims 26, 33, and 40 should be allowed for at least their dependence upon claims 22, 29, and 36, and for the specific limitations recited therein.

Applicants respectfully submit that the cited prior art fails to disclose or suggest all of the elements of the claimed invention. These distinctions are more than sufficient to render the claimed invention unobvious. It is therefore respectfully requested that all of claims 22-42 be allowed, and this application passed to issue.

If for any reason the Examiner determines that the application is not now in condition for allowance, it is respectfully requested that the Examiner contact, by telephone, the applicants' undersigned attorney at the indicated telephone number to arrange for an interview to expedite the disposition of this application.

In the event this paper is not being timely filed, the applicants respectfully petition for an appropriate extension of time. Any fees for such an extension together with any additional fees may be charged to Counsel's Deposit Account 50-2222.

Respectfully submitted,



Sejoon Ahn
Registration No. 58,959

Customer No. 32294
SQUIRE, SANDERS & DEMPSEY LLP
14TH Floor
8000 Towers Crescent Drive
Tysons Corner, Virginia 22182-2700
Telephone: 703-720-7800
Fax: 703-720-7802

SA:jf:ksh